ERDC MSRC/PET TR/00-11

Parallel Simulation of Flows in Open Channels at a Super-Critical Condition using the Finite Element Method

by

Shahrouz Aliabadi
Andrew Johnson
Bruce Zellars
Ade Abatan
Charlie Berger

28 March 2000

06h0112000

Parallel Simulation of Flows in Open Channels at a Super-Critical Condition Using the Finite Element Method

Shahrouz Aliabadi[1], Andrew Jonhson[2], Bruce Zellars[1], Ade Abatan[1], and Charlie Berger[3]

1 – Department of Engineering
Clark Atlanta University
223 James P. Brawley Dr. S. W.
Atlanta, Georgia 30314
Email: aliabadi@cau.edu

2 – Network Computing Services, Inc.
Army HPC Research Center
1200 Washington Ave. S.
Minneapolis, MN 55415
Email: ajohn@networkcs.com

3 – U.S. Army Engineer Research and Development Center
Coastal and Hydraulics Laboratory
CEERD-HE
3909 Halls Ferry Road
Vicksburg, MS 39180-6199
Email: berger@HL.wes.army.mil

1- Introduction

There has been much scientific work devoted to the design of channel transitions. Considerable experimental work has classified the behavior of flow in subcritical and supercritical conditions for both contraction and expansion channels [1,2]. The design of channels can be very different depending on the flow characteristics. The main parameter distinguishing such characteristics is the Froude number defined as

$$F_r = \sqrt{\frac{V_\infty^2}{Lg}}, \tag{1}$$

where $V_¥$ is the free-stream velocity, L is the free-stream water height and g is the gravitational acceleration. In subcritical flows where the Froude number is less than one, the effect of a transition is confined to the vicinity of the structure. Such channels are easier to analize both experimentally and theoretically. However, at a supercritical condition where the Froude number is greater than one, the existence of standing waves results in various surface elevations. These waves are formed near the contractions or expansion and are carried down stream for a significant distance. This makes the analysis of such problems much more difficult [1,2].

Computer modeling is a practical tool for studying flow behavior in open channels. In the past, due to hardware and software limitations, most of the models were developed only for simple 2D problems. Recent advancements in both hardware and software have

allowed us to routinely simulate 3D free-surface applications with complex geometries. Numerical simulations of free-surface flow applications are based on the solution of a complex set of partial differential equations governing the conservation of mass and momentum, commonly known as Navier-Stokes equations. In our approach, we use stabilized finite element methods to solve the governing equations. The Navier-Stokes equations are written over a discrete domain made of millions of small elements. These equations are integrated over each element, resulting in a set of coupled nonlinear equations which are solved iteratively to obtain the velocity, pressure and the location of the fluid interface over the discrete domain.

Generally, there are two distinct approaches in numerical simulation of free-surface applications. Depending on the physical characteristic of the problem, either a "moving mesh" or "fixed mesh" technique is used. In the moving mesh technique, the nodal coordinates on the free-surface are moved to track the motion of the free-surface [3-6]. The space-time finite element method is an example of the moving mesh technique [4,6]. In the space-time method, the finite element formulations are written over the space-time domain. As a result, the motion of the free-surface is taken into account automatically by simply moving the mesh. Here, the finite element functions are both linear in space and time, continuous in space, but discontinuous in time. This allows the solution of space-time slabs one at a time [3,7].

In moving mesh techniques, the computational mesh needs to change to account for the motion of the free-surface. In applications where the deformation of the interface is large, these moving techniques usually result in large element distortion. As the element distortion grows to an unacceptable limit, generation of a new mesh and projecting the solution from the old mesh to the new one is required. In complex 3D free-surface flow applications, this procedure is extremely difficult and time consuming (almost impossible). In such cases, computations over a fixed mesh is more desirable.

Most of the fixed mesh techniques are based on the volume of the fluid (VOF) approach [8]. In the VOF, the Navier-Stokes equations are solved over a non-moving mesh. A free-surface function with two distinct values serves as a marker identifying the location of the free-surface. This function is transported throughout the computational domain with a time-dependent advection equation [9-13]. Generally, the accuracy of fixed mesh techniques depends on how accurate the free-surface function is represented. Therefore, mesh resolution becomes a prime factor in determining the accuracy of these techniques [10-13].

Mass conservation is one of the main concerns in computations using fixed meshes. Numerical errors and algorithm instabilities usually result in fictitious mass transfer during the computations. This unphysical phenomenon will cause inaccuracies in long-term time-integration. Recently, we have designed a mass-conservation/interface-sharpening algorithm which eliminates this problem [9]. In this algorithm, our objective is to accurately represent and advect the free-surface function in such a way that the fluid mass is conserved over the time. Many test problems have been solved to prove the accuracy of this method [9-11].

In this paper, we explain the governing equations and finite element formulation in Sections 2 and 3. The mass-conservation/interface-sharpening algorithm is discussed in Section 4. In Sections 5 and 6 we present the iterative algorithm and parallel implementation strategy, respectively. The numerical simulations for a contraction channel at a supercritical condition are provided in Section 7. Finally, concluding remarks are provided in Section 8.

2- Governing Equations

Let us denote Ω and $(0,T)$ the space and time domains, respectively. The boundary of Ω is defined by Γ. Using this notation, the governing equations are written as:

$$r\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} - \mathbf{g}\right) - \nabla\cdot\sigma = 0 \qquad \text{on } \Omega \quad \forall t \in (0,T), \tag{2}$$

$$\nabla\cdot\mathbf{u} = 0 \qquad \text{on } \Omega \quad \forall t \in (0,T), \tag{3}$$

where

$$\sigma = -p\mathbf{I} + 2\mu\varepsilon(u), \tag{4}$$

$$\varepsilon = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^{T}). \tag{5}$$

Here $\mathbf{u}$, p, r, $\mathbf{g}$, and μ are the velocity, pressure, density, gravitational force, and dynamic viscosity, respectively. The strain tensor is denoted by ε, and $\mathbf{I}$ represents the identity tensor. Equations (2-3) are completed by an appropriate set of boundary and initial conditions:

$$\mathbf{u} = \mathbf{u}_s \qquad \text{on } \Gamma_u, \tag{6}$$

$$\mathbf{n}.\sigma = \mathbf{h} \qquad \text{on } \Gamma_{h_u}, \tag{7}$$

$$\mathbf{u}(t=0) = \mathbf{u}_0 \qquad \text{on } \Omega. \tag{8}$$

To simulate free-surface flow problems, we assume there are two fluids interacting with each other where the effect of one fluid on the other one is very small. For example, in water flows, air is constantly interacting with water. Since the density of air is almost 1000 times less than water, its effect on the water is very small.

The interface function f has two distinct values (0,1) and is used to differentiate between the two fluids. A time-dependent advection equation transports this function throughout the computational domain with the fluid velocity

$$\frac{\partial f}{\partial t} + \mathbf{u}\cdot\nabla f = 0 \qquad \text{on } \Omega \quad \forall t \in (0,T). \tag{9}$$

Using f, the density and viscosity can be calculate as:

$$r = fr_A + (1-f)r_B, \tag{10}$$

$$m = fm_A + (1-f)m_B, \tag{11}$$

where the subscripts *A* and *B* denote the fluid *A* and fluid *B*. Initially, f is set to 0 in fluid *A* and 1 in fluid *B*.

3- Finite Element Formulations

The finite element formulations are based on the SUPG (stabilized-upwind/Petrov-Galerkin) and PSPG (pressure-stabilized/Petrov-Galerkin) techniques. The SUPG term allows us to solve flow problems at high speeds and the PSPG term eliminate instabilities associated with the use of equal order interpolation functions for both pressure and velocity [9-11,14-16].

In the finite element formulation, we first define appropriate sets of trial solution spaces $S^h_{\mathbf{u}}$, S^h_p, and S^h_ϕ, and weighing function spaces $V^h_{\mathbf{u}}$, V^h_p and V^h_ϕ for the velocity, pressure, and the free-surface function, respectively. The stabilized finite element formulation of Equations (2,3,9) can then be written as follows: for all $\mathbf{w}^h \in V^h_{\mathbf{u}}$, $q^h \in V^h_p$, and $y^h \in V^h_\phi$, find $\mathbf{u}^h \in S^h_{\mathbf{u}}$, $p^h \in S^h_p$, and $f^h \in S^h_\phi$ such that:

$$\int_\Omega \mathbf{w}^h \cdot r[\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{g}]d\Omega + \int_\Omega \varepsilon(\mathbf{w}^h) : \sigma(p^h, \mathbf{u}^h) d\Omega$$
$$+ \int_\Omega q^h_p \nabla \cdot \mathbf{u}^h d\Omega + \int_\Omega y^h (\frac{\partial f^h}{\partial t} + \mathbf{u}^h \cdot \nabla f^h) d\Omega + \sum_{e=1}^{ne} \int_{\Omega^e} \tau_c \nabla \cdot \mathbf{w}^h \rho \nabla \cdot \mathbf{u}^h d\Omega$$
$$+ \sum_{e=1}^{ne} \int_{\Omega^e} \frac{\tau_m}{\rho} \left[\mathbf{u}^h \cdot \nabla \mathbf{w}^h - \nabla \cdot \sigma(q^h_p, \mathbf{w}^h)\right] \cdot \left[\rho[\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h - \mathbf{g}] - \nabla \cdot \sigma(p^h, \mathbf{u}^h) \right] d\Omega$$
$$+ \sum_{e=1}^{ne} \int_{\Omega^e} \tau_i \nabla y^h \cdot \nabla f^h d\Omega = \int_{\Gamma_{h_u}} \mathbf{w}^h \cdot \mathbf{h}\, d\Gamma . \qquad (12)$$

The parameters τ_c, τ_m, and τ_i are the stabilization parameters:

$$\tau_m = \left[\left(\frac{2\|\mathbf{u}\|}{h} \right)^2 + \left(\frac{4\mu}{\rho h^2} \right)^2 \right], \qquad (13)$$

$$\tau_c = \frac{h}{2\|\mathbf{u}\|} z, \qquad z = \begin{cases} R_u / 3 & R_u \le 3 \\ 1 & 3 < R_u \end{cases}, \qquad (14)$$

$$\tau_i = \frac{h}{2\|\mathbf{u}\|}, \qquad (15)$$

where h is the element length and R_u is the cell Reynolds number (see [4,6,9] for details).

In Equation (12), the first four integrals together with the right-hand-side represent the Galerkin formulation of the governing equations. The first, second and third series of element-level integrals in the formulation are the least-square stabilization of the continuity equation, the SUPG stabilization for momentum equations, and the artificial diffusion stabilization for the interface function, respectively. The diffusion formulation for the interface function eliminates the numerical undershoots (below 0) and overshoots

(above 1) of the interface function around the interface. In the mass-conservation/interface-sharpening algorithm, we recover the sharpness of the interface in such a way that the global mass conservation for each fluid is enforced.

4- Mass-Conservation/Interface-Sharpening

The finite element formulation in Equation (12) introduces numerical diffusion for the interface function f. To recover the sharpness of the interface function, after each iteration, f is replaced be f_{new} as follows:

$$f_{new} = \begin{cases} b^{(a-1)} f^a & 0 \le f \le b \\ 1-(1-b)^{(a-1)}(1-f)^a & b \le f \le 1 \end{cases}, \tag{16}$$

where $1.2 \le \alpha \le 1.5$ is a sharpening parameter [9], and $0 \le \beta \le 1$ is a point satisfying the global conservation of mass for each fluid.

To determine b, we satisfy the mass conservation at a given time t for f_{new}. Therefore:

$$r_A \int_\Omega f_{new} d\Omega = m_A + r_A \int_t \int_\Gamma f \mathbf{u}.\mathbf{n}\, d\Gamma dt\,, \tag{17}$$

$$r_B \int_\Omega f_{new} d\Omega = m_B + r_B \int_t \int_\Gamma (1-f) \mathbf{u}.\mathbf{n}\, d\Gamma dt\,, \tag{18}$$

where m_A and m_B are the initial mass of Fluid A and Fluid B, respectively. Note that we only need to satisfy either of the Equations (17) or (18). Combining Equations (16) and (17) and assuming that the parameter α is given and constant, results in

$$Mb^{(1-a)} + N(1-b)^{(1-a)} = R\,, \tag{19}$$

where M, N, and R are all function of b. This nonlinear equation is solved using a Newton-Raphson algorithm. Typically, with the initial guess of 0.5, the algorithm converges in three iterations [9,17-18].

5- Iterative Solution Strategy

The discretization of the finite element formulation results in a series of coupled, nonlinear systems of equations that need to be solved at every time step. For 3D complex problems, these systems include millions of unknowns, and the solution of these equations needs special attention. We solve these nonlinear systems iteratively using the GMRES algorithm [19]. For very large systems of equations, we use a matrix-free iteration strategy to obtain the solution of the nonlinear system. This element-vector-based computation totally eliminated the need to form any matrices, even at the element level [4,20].

6- Parallel Implementation

The computation of 3D free-surface flow applications are of very large-scale. Parallel supercomputers with hundreds of fast processors, such as CRAY T3E and IBM SP, are used to reduce the computational time. In the parallel implementation, we use a message-passing computing paradigm which makes the inter-processor communication explicit. This is accomplished using the MPI (message passing interface) libraries. Prior to the

computation, the finite element mesh is partitioned into contiguous subdomains, and these subdomains are assigned to individual processors. To ensure proper load balancing on each processor, each subdomain contains approximately the same number of elements.

The element-level computations are carried out independently for each processor. Data transfers between the elements and nodes is accomplished in two steps. First, data is gathered or scattered between the elements and the nodes residing on the same processor. This step does not involve any communication. In the second step, the gather and scatter operations are performed to exchange data across the processors only for those nodes residing on the boundary of subdomains. The two-steps gather and scatter operation leads to a more efficient communication strategy. Figure 1 shows the linear scalability of the algorithm on the Cray T3E. As the number of processors increases to 128, the speedup factor is 110. Clearly, the communication cost increases as the number of the processors increases, but for practical problems, the communication cost remains around 5 percent of the overall CPU time. A finite element mesh with one million elements is used to generate the results in Figure 1.

7- Application to Contraction Channel

The contraction channel walls are composed of two equal circular arcs each having a radius of 75 inch (see Figure 2). Water at high velocity enters the 24 inch wide channel at the speed of 85.3 inch/s and passes the narrow section of the channel which is 12 inches wide. The Froude number with respect to the entering water elevation of 1 3/16 inch is 4.0. The 3D geometry of the channel is shown in Figure 3.

We computed a solution to this problem using two different meshes. We compare the computed results with experimental data reported in [1].

Coarse Mesh Solution

The finite element mesh is made of 354,361 nodes and 336,000 hexahedral elements. The number of elements in the axial, vertical and cross-flow directions are 161, 31 and 71, respectively. The channel lengths before and after the contraction section are 19 11/16 inch and 96 ½ inch, respectively. Figure 4 shows the surface discretization of the finite element mesh.

The numerical computation starts with an initial uniform velocity field equal to the entering water velocity. The time step is set to $t = 0.00346$ s. As computations continue, water waves are formed in the contraction section and reflected back into the narrow section of the channel from the channel walls.

Figure 5 shows the comparison of the computed water elevation along the channel wall with experiment. All the flow features observed from the experiments are also captured in the computation. Figure 6 shows the comparison of the computed water elevation along the channel center with experiment. In these two figures, the computed results are very comparable with the experimental data.

Figures 7 and 8 show the pressure distribution on the channel surface and the surface of the water in the channel at t = 2.56 s, respectively. As in the experimental observations, the waves are reflected into the channel forming a "x" pattern.

Fine Mesh Solution

In every numerical method, a more refined mesh should yield in a better solution. In our case, we strongly believe that mesh refinement plays a significant role in capturing the details in free-surface flow problems accurately. In theory, the interface function should be a discontinuous function, and as we increase the level of refinement, we increase the accuracy in representing the interface function closer to what it should be. To justify this argument, we also computed the previous problem with a more refined finite element mesh.

Here, we use a mesh consisting of 2,715,171 nodes and 2,640,000 hexahedral elements. The number of elements in the axial, vertical and cross-flow directions are 401, 61 and 111, respectively. The channel lengths before and after the contraction section are 19 11/16 inch and 136 inch, respectively. Note that this mesh has a longer narrow section than in the coarse mesh. As was the case in the coarse mesh computation, the numerical computations start with the initial uniform velocity equal to the entering water velocity. The time step is also set to t = 0.00346 s.

Figure 9 and 10 shows the comparison of the computed water elevation along the channel wall and the channel center with experiment, respectively. In this case, the comparison is more favorable than was the case with the coarse mesh solution. The surface of the water in the channel at t = 2.56 s is shown in Figure 11.

8- Concluding Remarks

We developed a highly efficent and accurate finite element method for free-surface flow problems. The finite element formulations have been implemented in parallel using MPI libraries. The linear scalability of the solver was demonstrated. We applied this finite element flow solver to a complex 3D application involving water flow throught channels at supercritical conditions. We demonstrated the accuracy of the computations by comparing the results with experimental data. The computations were carried out using a coarse and a fine mesh. As expected, the fine mesh solution is superior to that of the coarse mesh solution.

Acknowledgment/Disclaimer

We thank the DoD High Performance Computing Modernization Program ERDC Major Shared Resource Center through Programming Environment and Training (PET) for funding this work. We also thank the Army HPC Research Center for partial support and providing computer time. Views, opinions, and/or findings contained in this report are those of the authors and should not be constructed as an official Department of Defense position, policy, or decision unless so designated by other official documentation.

REFERENCES

[1] T. Ippen, M. Asce and John H. Dawson, "**Design of Channel Contraptions,**" *High Velocity Flow*, 326-346, 1951.

[2] H. Rouse, M. Asce, B. Bhoota, "**Design of Channel Expansions,**" *High Velocity Flow*, 347-363, 1951.

[3] A. Johnson and T. Tezduyar, "**Mesh Update Strategies in Parallel Finite Element Computations of Flow Problems with Moving Boundaries and Interfaces,**" *Computer Methods in Applied Mechanics and Engineering*, 119:73-94, 1994.

[4] S. K. Aliabadi and T.E. Tezduyar, "**Space-time Finite Element Computation of Compressible Flows Involving Moving Boundaries and Interfaces,**" *Computer Methods in Applied Mechanics and Engineering*, **107** (1993), 209-223.

[5] S. K. Aliabadi and T. E. Tezduyar, "**Parallel Fluid Dynamics Computations in Aerospace Applications,**" *International Journal for Numerical Methods in Fluids*, **21** (1995), 783-805.

[6] M. Behr and T. Tezduyar, "**Finite Element Solution Strategies for Large-Scale Flow Simulations,**" *Computer Methods in Applied Mechanics and Engineering*, **112** (1994), 3-24.

[7] A.A. Johnson and T.E. Tezduyar, "**Parallel Computation of Incompressible Flows with Complex Geometries,**" International Journal for Numerical Methods in Fluids, **24** (1997), 1321-1340.

[8] C. W. Hirt and B. D. Nichols, "**Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries,**" *Journal of Computational Physics*, 39:201-225, 1981.

[9] S. Aliabadi and T. Tezduyar, "**Stabilized-Finite-Element/Interface-Capturing Technique for Parallel Computation of Unsteady Flows with Interfaces,**" to appear in *Computer Methods in Applied Mechanics and Engineering*.

[10] S. Aliabadi and T. Tezduyar, "**Simulation of Two-Fluid Interactions Using Parallel Finite Element Method,**" *Abstract of the Fourth Mississippi State Conference on Differential Equations and Computational Simulations*, Mississippi State University and Electronic Journal of Differential Equations, Starkville, Mississippi, May 1999, 21 - 22.

[11] S. Aliabadi and T. Tezduyar, "**Parallel Finite Element Simulation of Two-Fluid Flow Applications,**" *Abstracts of the Fifth U.S. National Congress on Computational Mechanics*, University of Colorado, Boulder, Colorado, August 1999.

[12] T. Tezduyar, S. Aliabadi and M. Behr, "**Enhanced-Discretization Interface-Capturing Technique (EDICT) for Computation of Unsteady Flows with Interfaces,**" *Computer Methods in Applied Mechanics and Engineering*, **155** (1998), 235-248.

[13] T. Tezduyar and S. Aliabadi, "**EDICT for 3D Computation of Two-Fluid Interfaces,**" to appear in *Computer Methods in Applied Mechanics and Engineering.*

[14] S. K. Aliabadi, S. E. Ray and T. E. Tezduyar, "**SUPG Finite Element Computation of Viscous Compressible Flows Based on the Conservation and Entropy Variables Formulations,**" *Computational Mechanics*, **11** (1993), 300-312.

[15] T. E. Tezduyar, S. K. Aliabadi, M. Behr and S. Mittal, "**Massively Parallel Finite Element Simulation of Compressible and Incompressible Flows,**" *Computer Methods in Applied Mechanics and Engineering*, **119** (1994), 157-177.

[16] T. E. Tezduyar, S. K. Aliabadi, M. Behr, A. Johnson and S. Mittal, "**Parallel Finite Element Computation of 3D Flows,**" *IEEE Computer*, (1993) 27--36.

[17] S. Aliabadi and T. Tezduyar, "**3D Simulation of Two-Fluid and Free-Surface Flows with the Stabilized-Finite-Element/Interface-Capturing Method,**" Modeling and Simulation Based Engineering (eds. S. Atluri and P. O'Donoghue), *Proceeding of International Conference on Computational Engineering Science*, Atlanta, Georgia (1998).

[18] S. Aliabadi, K. Shujaee and T. Tezduyar, "**Parallel Simulation of Two-Phase Flow Problems Using the Finite Element Method,**" *Proceeding of The 7th Symposium on the Frontiers of Massively Parallel Computation (Frontiers'99),* Annapolis, Maryland, February 1999, 21-25.

[19] Y. Saad and M. Schultz**,** "**GMRES: Generalized Minimal Residual Algorithm for Solving Nonsymmetic linear Systems,**" *SIAM Journal of Scientific and Statistical Computing*, **7** (1986), 856-896.

[20] Z. Johan, T.J.R. Hughes and F. Shakib, "**A Globally Convergent Matrix-free Algorithm for Implicit Time-marching Schemes Arising in Finite Element Analysis in Fluids,**" *Computer Methods in Applied Mechanics and Engineering*, **87** (1991), 281-304.

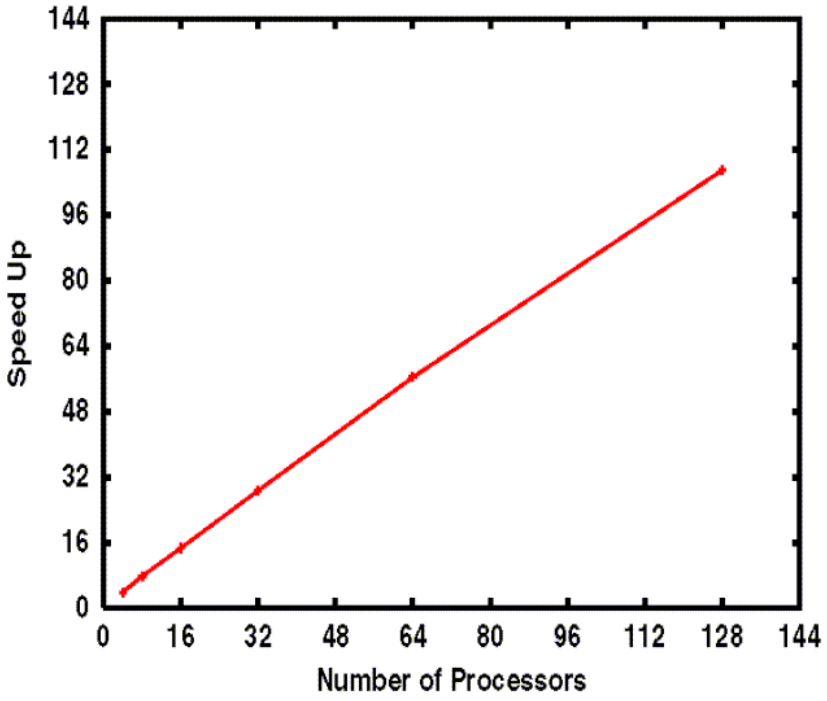

Figure 1. Speed up in parallel computation on the Cray T3E.

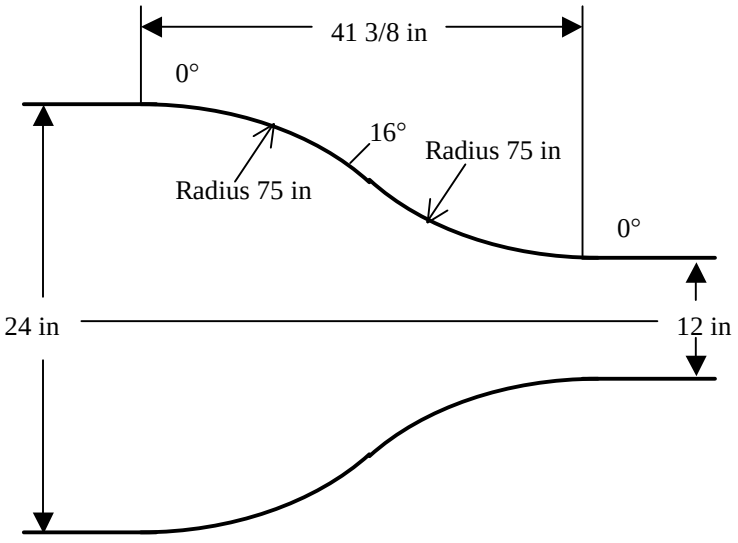

Figure 2. Contraction channel.

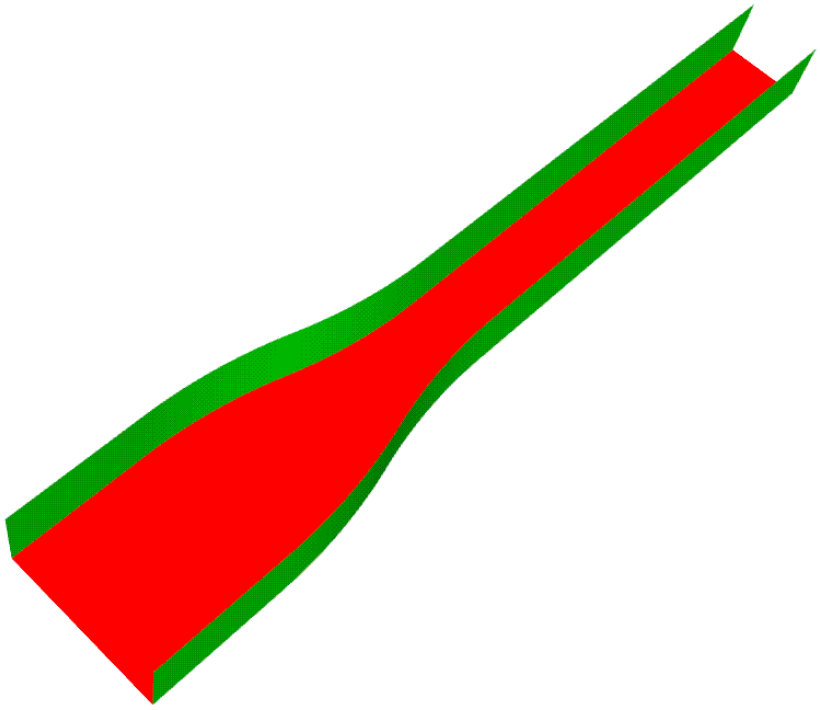

Figure 3. Contraction channel geometry.

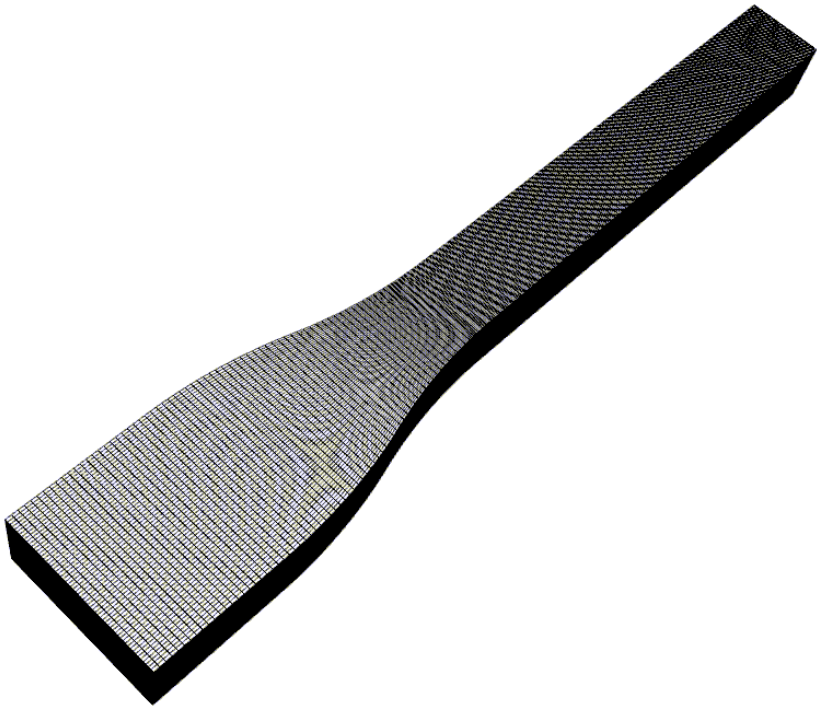

Figure 4. Coarse Mesh Solution. Surface discretization of the finite element mesh.

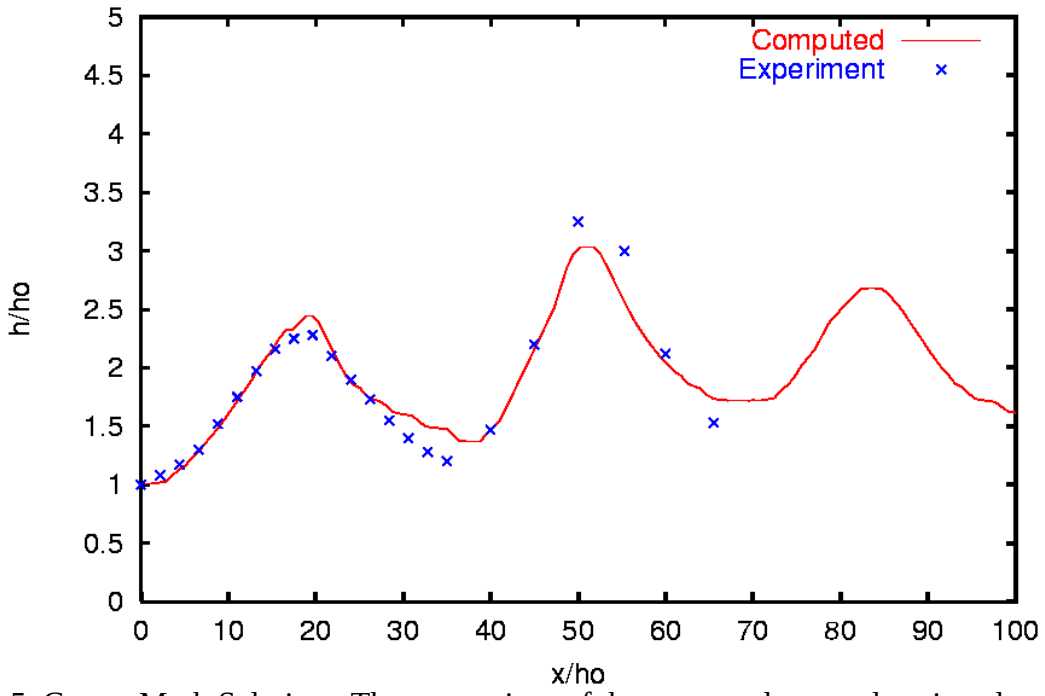

Figure 5. Coarse Mesh Solution. The comparison of the computed water elevation along the channel wall with experiment.

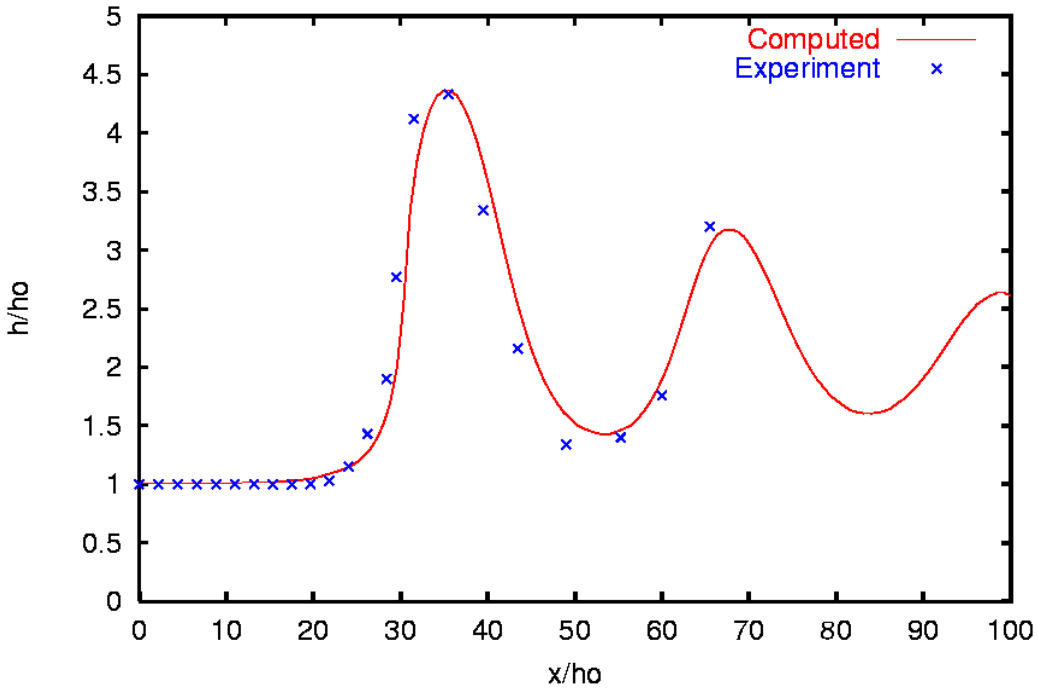

Figure 6. Coarse Mesh Solution. The comparison of the computed water elevation along the channel center with experiment.

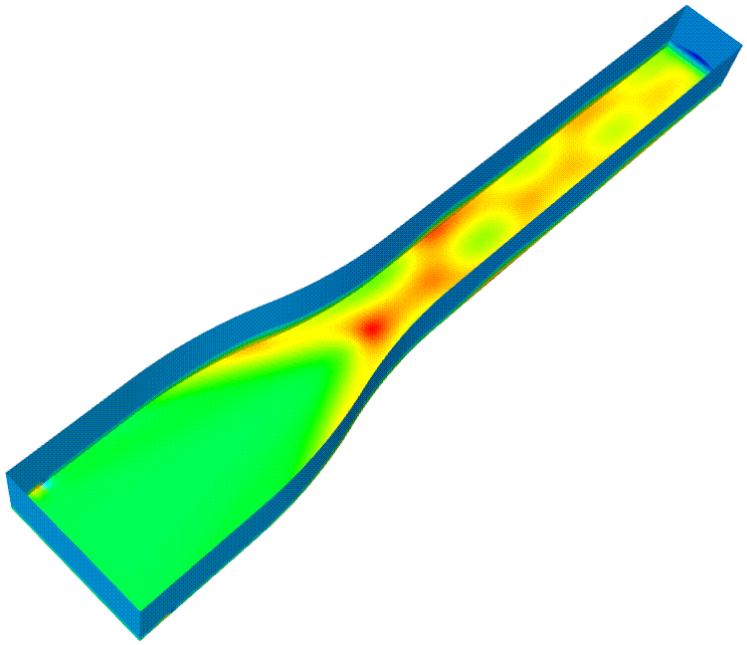

Figure 7. Coarse Mesh Solution. Pressure distribution on the channel surface at t = 2.56 s.

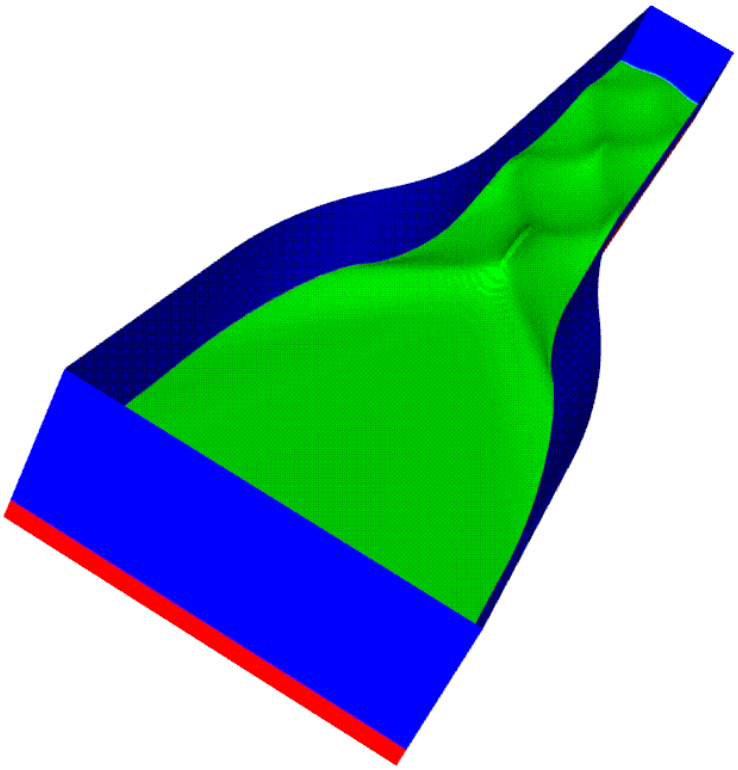

Figure 8. Coarse Mesh Solution. Surface of the water in the channel at t = 2.56 s.

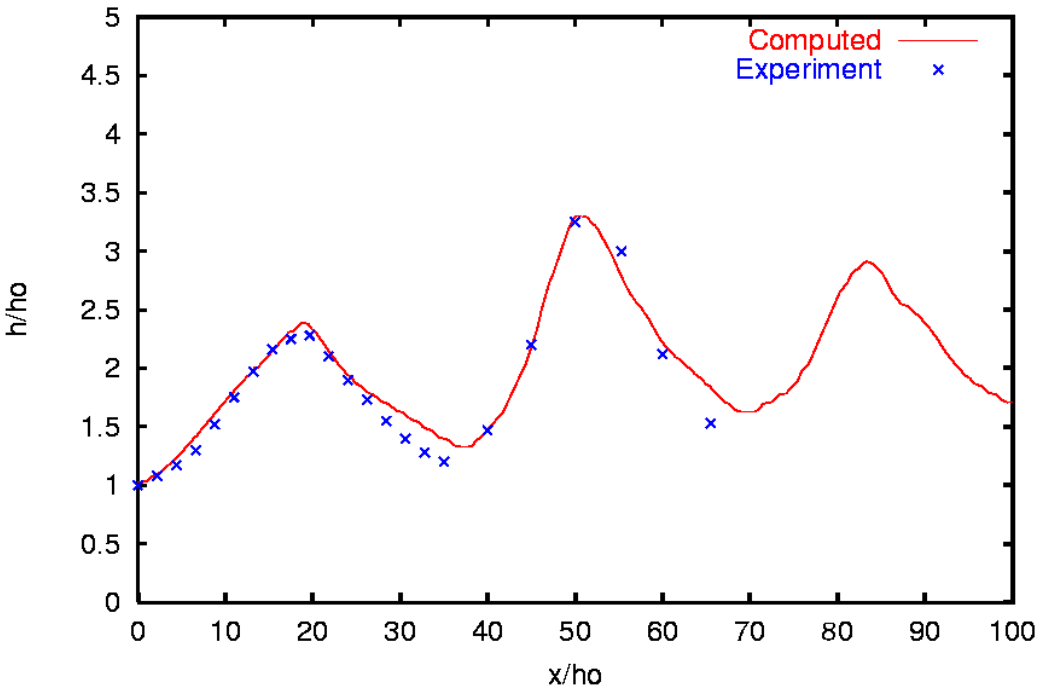

Figure 9. Fine Mesh Solution. The comparison of the computed water elevation along the channel wall with experiment.

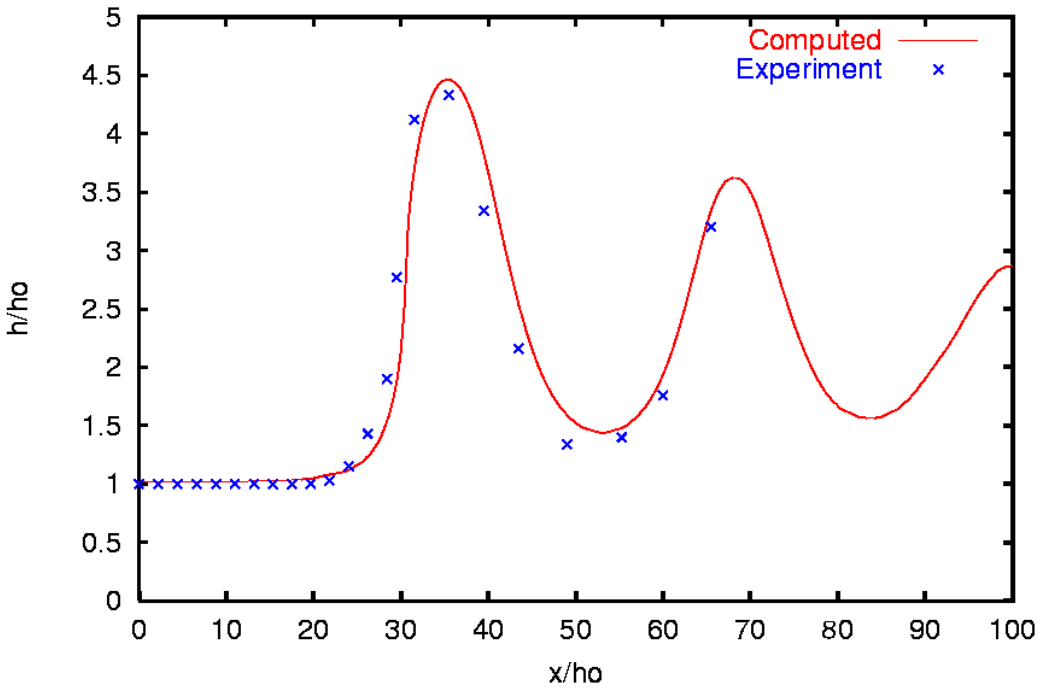

Figure 10. Fine Mesh Solution. The comparison of the computed water elevation along the channel center with experiment.

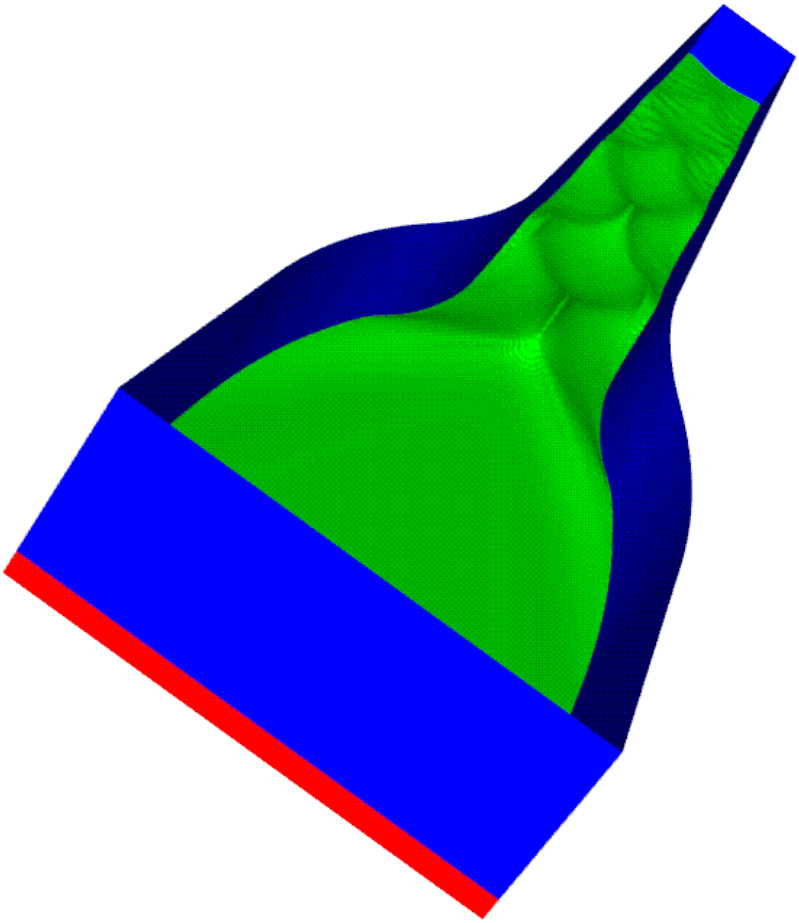

Figure 11. Fine Mesh Solution. Surface of the water in the channel at t = 2.56 s.